REMARKS

I. Introduction

In response to the Office Action dated February 27, 2003, the claims have not been amended. Claims 1-12 remain in the application. Re-examination and re-consideration of the application, as amended, is requested.

II. Prior Art Rejections

In paragraphs (1)-(2) of the Office Action, claims 1, 3-5, 7-9, and 11-12 were rejected under 35 U.S.C. §103(a) as being unpatentable over Kish et al., U.S. Patent No. 5,890,176 (Kish) in view of Gerard et al., U.S. Patent No. 5,974,428 (Gerard). In paragraph (6) of the Office Action, claims 2, 6, and 10 were rejected under 35 U.S.C. §103(a) as being unpatentable over Kish in view of Gerard as applied to claims 2, 6, and 10, and further in view of Cohen et al., U.S. Patent No. 6,324,543 B1 (Cohen).

Specifically, claims 1, 5, and 9 were rejected as follows:

> As per claims 1, 5, and 9, Kish teaches "obtaining a request to store an object" at col. 2, lines 20-34;
>
> "determining if a requested file version is lower than an object introduction version of the object" at col. 2, lines 20-34, col. 10, lines 42-59;
>
> Moreover, Kish teaches "an object-oriented computer system having a memory and a data storage device coupled thereto" at col. 3, lines 8-37;
>
> Kish does not teach "streaming out a class of the object in the requested file version if the requested file version is equal to or higher than the object introduction version; and
>
> streaming out the class of the object in the object introduction version if the requested filed version is lower than the object introduction version". However, Gerard teaches these limitations at col. 8, lines 39-61.

Applicant traverses the above rejections for one or more of the following reasons:

(1) Neither Kish, Gerard, nor Cohen teach, disclose or suggest streaming out a class of an object;

(2) Neither Kish, Gerard, nor Cohen teach, disclose or suggest streaming out a class in a requested file version if the requested file version is equal to or higher than the object introduction version; and

(3) Neither Kish, Gerard, nor Cohen teach, disclose or suggest streaming out the class of the object in the object introduction version if the requested file version is lower than the object introduction version.

Independent claims 1, 5, and 9 are generally directed to storing object data in a particular version. Specifically, when an object is being stored, the invention provides for storing a particular version of the object. More specifically, the version of the file (that the object is being stored in) is compared to the version of the object when the object was introduced/originated. If the file version is the same or newer than when the object was first introduced, the object is streamed out (e.g., stored) in the file version. However, if the file version is older than when the object was introduced, the object is streamed out (e.g., stored) in the object introduction version. The cited references do not teach nor suggest these various elements of Applicant's independent claims.

The Office Action admits that Kish fails to teach the streaming steps. In this regard, Gerard is relied upon to teach these claimed elements. However, contrary to the assertion in the Office Action, Gerard fails to teach these claimed limitations.

In Gerard, a Java virtual machine uses a VersionLoader object which loads class files. The VersionLoader is passed a name of a desired class and returns a newly loaded class (see col. 5, lines 21-32). As illustrated in FIG. 4, when a request for a new object is received, a class must be loaded to support that request (see col. 8, lines 39-43). The request is sent to the VersionLoader object which utilizes a mapping mechanism (see col. 8, lines 43-47). The mapping mechanism supports a versioning policy that determines which class to return (see col. 8, lines 48-51).

The presently claimed invention provides for streaming out a particular version of an object. In this regard, the particular object that is streamed may be said to be based on a streaming policy (although Applicants disagree with referring to Applicant's invention as a "policy"). However, such an alleged particular policy in the present invention is clearly distinguishable and is not even remotely suggested by the policies described in Gerard.

None of Gerard's versioning policies teach, disclose, or suggest the presently claimed invention. In the present claims, an object introduction version is streamed out if the file version is lower than the object introduction version. Further, the object is streamed out in the file version if the file version is equal to or higher than the object introduction version. Gerard fails to describe any such policy. In col. 7, lines 10-15, Gerard describes a policy where if a user requests a certain specific version of a class, that particular version is provided. Additionally, if a default version of the class is specified, the most current production version of the class is provided (see col. 7, lines 15-

18). Accordingly, Gerard describes a policy that provides a particular version of a class if requested and otherwise provides the most recent version of the class. Gerard further describes the use of an abstract versioning policies in col. 7, lines 37-44. However, once again, Gerard fails to describe the particular streaming set forth in the present independent claims. Additionally, nowhere in Gerard is there any description of an object introduction version or a requested file version as claimed.

The Office Action relies on col. 8, lines 39-61 to teach the streaming out as claimed. However, col. 8, lines 39-61 merely describes the use of the VersionLoader object and a versioning policy that is supported by a version mapping mechanism used by the VersionLoader object. In this regard, the mere presence and use of an abstract versioning policy does not even remotely describe the use of a particular "policy". Nor does it describe streaming out an object in a particular version as claimed.

Further, Gerard merely provides a class to a user for loading. In this regard, Gerard does not describe the storage of an object by streaming out a class or a request to store an object. Storing/streaming out a class of an object is distinguishable from providing and loading a new object for a user.

In addition to the above, Gerard teaches away from the present invention. Specifically, in Gerard's background, cols. 2-3, lines 66-6, Gerard describes the problems with using a class hierarchy as utilized in the present invention. Gerard provides that the use of a class hierarchy is extremely inefficient and can measurably reduce system performance. As a result, Gerard, teaches a methodology that does not use such a class hierarchy. In this regard, Gerard utilizes a class versioning and mapping mechanism to cross reference a requested class, select the most recent or best version of the requested class, and return an object to he user that belongs to the appropriate class (see col. 3, lines 32-39). More specifically, Gerard describes the use of a naming system as illustrated in FIGS. 2-4 and described throughout the specification (see for e.g. col. 7, lines 33-37). Such a teaching teaches away from using the hierarchical class system and using the comparisons and streaming methodology as claimed.

Moreover, the various elements of Applicant's claimed invention together provide operational advantages over Kish, Gerard, and Cohen. In addition, Applicant's invention solves problems not recognized by Kish, Gerard, and Cohen.

Thus, Applicant submits that independent claims 1, 5, and 9 are allowable over Kish, Gerard, and Cohen. Further, dependent claims 2-4, 6-8, and 10-12 are submitted to be allowable over Kish, Gerard, and Cohen in the same manner, because they are dependent on independent claims 1, 5, and 9, respectively, and thus contain all the limitations of the independent claims. In addition, dependent claims 2-4, 6-8, and 10-12 recite additional novel elements not shown by Kish, Gerard, and Cohen.

III. Conclusion

In view of the above, it is submitted that this application is now in good order for allowance and such allowance is respectfully solicited. Should the Examiner believe minor matters still remain that can be resolved in a telephone interview, the Examiner is urged to call Applicant's undersigned attorney.

Respectfully submitted,

GATES & COOPER LLP
Attorneys for Applicant(s)

Howard Hughes Center
6701 Center Drive West, Suite 1050
Los Angeles, California 90045
(310) 641-8797

Date: May 23, 2003

By: ____________________
Name: Jason S. Feldmar
Reg. No.: 39,187

JSF/sjm

G&C 30566.90-US-01